Software Requirement Engineering

1. What is Trade-off Analysis?

Trade-off Analysis is the process of comparing different solutions in software development to make balanced decisions. It helps in selecting the best option considering benefits, costs, risks, and priorities.

---

2. Keys of Trade-off Analysis

1. **Defining Criteria –** Decide the factors for comparison, like cost, time, quality, performance.
2. **Assigning Weight –** Give importance to each criterion based on project needs.
3. **Evaluating Alternatives –** Check how each option meets the defined criteria.
4. **Analyzing Results –** Understand what you gain or lose with each option.
5. **Making Decision –** Choose the option that offers the best balance among criteria.

---

3. Techniques of Trade-off Analysis

1. **Decision Table –** Use a table to compare alternatives against criteria.
2. **MCDA (Multi-Criteria Decision Analysis) –** Score each option based on multiple criteria and their weights.
3. **ATAM (Architecture Trade-off Analysis Method) –** Focus on software architecture decisions and their impact on quality attributes.

4. Example of Trade-off Analysis

Scenario: Developing a mobile app.

- **Option A:** Fast delivery, higher cost.
- **Option B:** Lower cost, slower delivery.
- **Option C:** Medium cost, medium speed.

Trade-off Analysis:

- **Define criteria:** Cost, Time, User Experience
- **Assign weights:** Time (40%), Cost (30%), User Experience (30%)
- **Evaluate alternatives → Option C provides the best balance.**

1. What is Requirement Elicitation?

This is the process of gathering requirements from stakeholders (like users, customers, etc.). It involves techniques such as interviews, surveys, observations, and brainstorming sessions to understand what the users need.
Techniques include interviews, surveys, observations, and brainstorming to understand what users need.

Common Techniques of Requirement Elicitation (Detailed)

1. **Interview –** Directly talk to users, customers, or stakeholders.
   Helps in understanding their needs, expectations, and problems.
   Questions can be structured, unstructured, or semi-structured.
2. **Survey –** Collect information from a large group of people using questionnaires.

Useful for quantitative data and identifying common requirements.
Can be done online or offline for faster results.

3. **Workshop –** Organize group sessions with users and stakeholders.
   Encourages discussion, brainstorming, and agreement on requirements.
4. **Observation –** Watch how users interact with current systems or processes.
   Helps identify hidden needs or issues not mentioned in interviews.
   Can be direct or indirect observation.
5. **Prototyping –** Build a simple model of the system to show users.
   Users provide feedback, which helps refine requirements.

---

3. Importance of Requirement Elicitation

- **Reduce Cost –** Avoid mistakes and unnecessary features.
- **Fast Implementation –** Clear requirements help in quicker development.
- **Increase Bonus –** Satisfied users and successful projects improve rewards.

Risk Analysis & Impact Analysis

Risk Analysis

Definition:

Risk Analysis is the process of identifying, assessing, and managing risks in a project to reduce negative impact.

Purpose:

- Find potential problems before they occur.
- Plan actions to reduce risk.

Process:

- Identify risks → Assess probability and impact → Prioritize risks → Plan responses**.**

Outcome:

- Project is safer and better prepared for problems.

Impact Analysis

Definition:

Impact Analysis is the process of understanding the effects of changes in requirements or systems.

Purpose:

- Know how changes affect the system.
- Decide whether to accept changes.

Process:

- Identify change → Check affected areas → Assess impact → Plan actions.

Outcome:

- Changes are managed properly without breaking the system.

6 Differences between Risk Analysis and Impact Analysis

Risk Analysis	Impact Analysis
Focuses on potential problems	Focuses on effects of changes
Helps prevent risks	Helps manage changes
Looks at future uncertainties	Looks at current or proposed changes
Outcome is risk plan	Outcome is change plan

Risk Analysis	Impact Analysis
Deals with probability & severity	Deals with affected areas & impact
Helps reduce project failure	Helps maintain system stability

Analysis Quality Requirements

Definition:

It is refers to the characteristics that make requirements documents effective such as complete , error free, consistence and testable.

Characteristics of Analysis of Quality Requirements Management (R.M.)

1. **Complete** – All requirements are fully defined without missing details.
2. **Error Free** – No mistakes or wrong information in requirements.
3. **Consistent** – Requirements do not conflict with each other.
4. **Testable** – Each requirement can be checked and verified by testing.
5. **Measurable** – Progress and success can be measured clearly.
6. **Actionable** – Requirements can be practically implemented.
7. **Documented** – Requirements are properly written and recorded.

Importance of Analysis of Quality Requirements

- **Reduce Project Cost** – Clear requirements prevent rework and waste.
- **Improved Communication** – Everyone understands the same needs.
- **Provide Basis of Testing** – Testing is easier when requirements are clear.
- **Ensure Stakeholder Satisfaction** – Final product matches user expectations.

Interaction Between R.M and Architecture

- ## Definition
- Requirements Management (R.M.) and Software Architecture are closely connected in software engineering.
  R.M. ensures that all user needs and requirements are collected, documented, and managed properly.

- ## Interaction between R.M. and Architecture
- Requirements guide the architecture by defining what the system must do.
  Architecture responds by providing the structure that fulfills those requirements.
  They interact mainly through two activities: **Requirements Elicitation** and **Requirements Analysis**.
  R.M. collects and analyzes requirements, while architecture translates them into system design.
  Together, they ensure that the software is both **usable** and **technically sound**.

Architecture Design

- **Respond to Requirement** – Architecture is shaped according to functional and non-functional requirements. It ensures performance, security, scalability, and maintainability.
- **ASRS (Architecture Software Requirement Specification)** – Acts as a bridge that connects documented requirements with architectural design, ensuring no requirement is missed.

Introduction to Management

Definition:
It is a process of planning, organizing, leading and controlling an organization's resources to achieve its goals effectively and efficiently.

Keys of Management

1. **Controlling** – Monitoring activities, finding gaps, and correcting them to keep work on track.
2. **Leading** – Guiding, motivating, and influencing people to achieve organizational goals.
3. **Organizing** – Arranging resources, tasks, and people in a structured way.
4. **Setting Goals and Achieving Them** – Defining clear objectives and making efforts to fulfill them.

Importance of Management

- **Achieve Goals** – Helps the organization reach its objectives successfully.
- **Cost Minimization** – Reduces wastage and manages resources in a cost-effective way.
- **Efficient Resource Utilization** – Uses time, money, and manpower properly.
- **Organizational Growth** – Ensures development and long-term success of the organization.
- **Create Positive Work Environment** – Builds teamwork, motivation, and creativity among employees.

Sources of Requirement Elicitation

1. **Stakeholders** – Users, customers, managers, and employees who directly use or affect the system.
2. **Existing Documents** – Old reports, manuals, policies, and system specifications.
3. **Business Process** – Current workflows and operations of the organization.
4. **Market Study** – Competitor products and industry research.
5. **Regulations & Standards** – Government rules, legal requirements, and quality standards.

6. **System Interfaces** – Existing systems or software that need to connect with the new system.

UML (Unified Modeling Language)

Introduction

UML stands for **Unified Modeling Language**.
It is a **visual language** used in software engineering to design, understand, and document a system.
UML provides a **standard way** to show how a system works, how different parts connect, and how data flows.

Purpose of UML

- To **visualize** the structure and behavior of a system.
- To **communicate ideas** between developers, designers, and stakeholders.
- To **analyze and design** software before actual coding.
- To **reduce complexity** by breaking the system into diagrams.

Types of UML Diagrams

1. Structural Diagrams (Show static structure of the system)

- **Class Diagram** – Shows classes and relationships.
- **Object Diagram** – Shows objects at a particular time.
- **Component Diagram** – Represents software components.
- **Deployment Diagram** – Shows hardware and software deployment.

2. Behavioral Diagrams (Show dynamic behavior of the system)

- **Use Case Diagram** – Shows interactions between users and the system.

- **Sequence Diagram** – Shows how objects interact step by step.
- **Activity Diagram** – Shows workflows or processes.
- **State Diagram** – Shows states of an object and transitions.

Importance of UML

- Provides **clear understanding** of system design.
- Acts as a **blueprint** for developers.
- **Improves communication** between team and stakeholders.
- Helps in **early error detection** and reduces project cost.

Supplier Organization

Definition:

A **Supplier Organization** is a company or group that **provides goods or services** to other organizations (clients) to help them achieve their goals. It plays an important role in the **supply chain**, making sure materials or services are delivered **on time, efficiently, and properly**.

Detailed Notes on Supplier Organization

1. **Self Organization** –
   Supplier organizations manage themselves, plan their work, and organize resources to meet client needs.
2. **Providing Goals or Services** –
   They provide goods or services according to client goals, making sure they meet quality standards.
3. **Bidding and Contracting** –
   Suppliers take part in **bidding** and **sign contracts** with clients to provide products or services at agreed terms.
4. **Supply Chain Role** –
   They are a **key part of the supply chain**, ensuring smooth flow of materials, services, and information.

5. **Supplier Management** –
   Organizations **monitor suppliers** for quality, reliability, performance, and timeliness.
6. **Organization Structure** –
   Suppliers have structured teams like **procurement, production, logistics, quality, and admin** to manage work efficiently.

Product Organization

Definition:

A **Product Organization** is a company or group that **creates, develops, and delivers products** to customers.
It focuses on **designing, producing, and improving products** to meet market and user needs effectively.

Detailed Notes on Product Organization

1. **Self Organization** –
   Product organizations manage themselves, plan their operations, and organize teams to develop products efficiently.
2. **Providing Goals or Services** –
   They deliver products according to customer requirements and ensure they meet **quality and usability standards**.
3. **Design and Development** –
   Responsible for **designing, prototyping, testing, and improving** products before release.
4. **Market Role** –
   Understands customer needs, market trends, and competition to create products that succeed in the market.
5. **Product Management** –
   Teams **monitor product performance, gather feedback, and make improvements** to maintain product quality and relevance.
6. **Organization Structure** –
   Typically structured into **R&D, design, production, marketing, and sales teams** for smooth product management.